Associate Professor Jujie WANG, PhD (Corresponding author)
E-mail: wjj0913@126.com
School of Management Science and Engineering, Nanjing University of Information Science and Technology, Nanjing, China
Student Chunchen FENG
E-mail: wjk1831346985@126.com
Changwang School of Honors, Nanjing University of Information Science and Technology, Nanjing, China
Student Junjie HE
E-mail: ycdthjj@163.com
School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
Student Liu FENG
E-mail: feng_liu_official@163.com
Student Yang LI
E-mail: ly1016080314@163.com
Changwang School of Honors, Nanjing University of Information Science and Technology, Nanjing, China

A NOVEL MULTI-FACTOR STOCK INDEX PREDICTION APPROACH USING PRINCIPAL COMPONENT ANALYSIS, FEATURE CLASSIFICATION AND TWO-STAGE LONG SHORT-TERM MEMORY NETWORK WITH RESIDUAL CORRECTION

Abstract. *Forecasting stock price index effectively has a great significance for investors to avoid risks and increase returns. In the study, a novel multi-factor prediction approach which integrates principal component analysis and feature classification into two-stage long short-term memory network with residual error correction, has been proposed for enhancing the prediction accuracy of stock price index. The principal component analysis is utilized to extract these key interrelated factors and generate a new data structure. The feature classification is developed to divide the data structure into several different datasets for facilitating the construction of the prediction model. The two-stage long short-term memory network based on residual error correction is established to capture the characteristics of stock market fluctuations and enhance the prediction accuracy. A series of comparative experiments from several benchmark models are provided to verify the performance of the developed model. The experiment results indicate that the developed model has the best prediction performance.*

Keywords*: Stock index prediction, Principal component analysis, Feature classification, Long short-term memory network, Residual error correction.*

JEL Classification: O30

DOI: 10.24818/18423264/54.3.20.06

1. Introduction

With the increasing number of people participating in stock investment activities, investors are in urgent need of an effective stock price prediction technique to help them maximize return and minimize loss (Roy et al., 2020). However, due to the nonlinearity and complexity of stock market fluctuations, it is an important and challenging task for forecasting stock price accurately.

Various methods have been conducted to predict the stock market recently. The current forecasting methods can be mainly divided into two categories including traditional statistical methods and machine learning (Liu, 2018; Baffour et al., 2019). These statistical methods have been proved by practice that they cannot get satisfactory prediction results of stock market due to the assumption of linearity and normality for samples data.

In recent years, with the development of emerging technologies, machine learning models have been developed to overcome the deficiencies of traditional statistical methods and have been widely applied in stock market prediction issues (Lee et al., 2019; Khashei and Hajirahimi, 2019). However, the shortcomings of the shallow neural network represented by back propagation (BP) cannot be ignored. They are prone to fall into the local optimal solution in the training process and cannot get the global optimal solution. In order to make up for the deficiency of BP neural network, some scholars put forward the research method of support vector machine (SVM) (Vilela et al., 2019). As a pattern recognition technology, SVM can break through the assumptions of financial models in a certain sense (Gong et al., 2019; Lahmiri, 2018). However, the promotion ability of the SVM is weak and the predicted result is not ideal due to the difficulty of determining the parameters of kernel function.

To overcome the deficiencies of the common machine learning models, the deep learning represented by recurrent neural network (RNN) is widely used in various prediction aspects (Xu et al., 2019). However, due to the deficiencies of the RNN including vanishing gradient and gradient explosion with long data, LSTM has been specially designed to handle these problems (Liu, 2019). LSTM is characterized by the use of memory blocks instead of ordinary hidden layer nodes. This characteristic is useful to ensure that information is stored across arbitrary delays and error signals are returned to points in time long ago. Thus, the problem of RNN can be solved. Based on the above advantages, LSTM model is widely used in many prediction fields (Sagheer and Kotb, 2019; Zhang et al. 2019). Considering the advantages of LSTM model, in this study, LSTM model is selected as a basic model to improve the prediction accuracy of stock index.

In addition to the choice of prediction model, it is also very important to rationalize the original data (Rufino et al., 2019). In the current research, most scholars build a one-dimensional data model through a single time series (Henrique et al., 2018). Some researchers realise that factors affecting stock price are not unique. Thus, they consider various influencing factors to improve the prediction accuracy (Zhong and Enke, 2017). Since there is a certain correlation between

DOI: 10.24818/18423264/54.3.20.06

various influencing factors, the additional noise will be introduced if the unprocessed multi-factor data group is directly input into the model for predictive analysis. In order to solve the above problems, Principal component analysis (PCA) is often used to extract the main features from high-dimensional data space composed of various factors, so as to improve the prediction accuracy (Wang and Wang, 2015). By using PCA, the redundancy between data can be avoided. Also, the dimension of data input can be reduced, and the operating efficiency of the algorithm can be improved.

Moreover, PCA alone is far from enough to process data considering the complexity of stock price changes. Hence, more accurate prediction results can be obtained by classifying the PCA data according to the characteristics of the data. Each data subset is modelled and analysed one by one. This approach takes full account of the inherent connection of the data which can gain better prediction. Besides, the prediction model with residual correction is more effective (Ouyang et al. 2019). The error range between the real value and forecast value can be analysed, and then the general accuracy of the proposed model can be improved by applying residual correction.

Synthesizing the above statement, a novel hybrid model composed of PCA, feature classification and two-stage LSTM, is developed to improve the prediction accuracy of stock price. Firstly, PCA is used to conduct data pre-processing and extract the main features from various factors. Secondly, a feature classification method based on growth rate is proposed to divide the data structure into several different datasets for facilitating the construction of the prediction model. Thirdly, in the two-stage LSTM model, the first stage LSTM is used to forecast the stock price. The second stage LSTM is used to conduct residual correction. To verify the validity of the model, data from S&P 500 are used as samples to explain the accuracy of the model. Also, a series of comparative experiments illustrate the results better.

There are four innovations in this paper. Firstly, multi-factor data are processed by PCA. In this way, the data can be considered comprehensively instead of simply importing multi-factor data into model. Secondly, classifying the data according to the growth rate of the valuation is an innovation. Considering the impact of inflation and other factors on stock prices, it is necessary and scientific to classify the data according to the growth rate of stock prices. Thirdly, LSTM is used in stock prediction. Compared with the traditional neural network, RNN proposed the concept of hidden state to extract the data features of sequence shape, and generated new sequences according to the classification. In addition, LSTM is a good solution to the problem of disappearing gradient of RNN. Finally, residual correction plays an essential role in improving the prediction accuracy of the model. The gap between predicted value and real value is narrated by residual correction, making the prediction of the model more fitting.

DOI: 10.24818/18423264/54.3.20.06

The remainder of this paper consists four parts. Section 2 briefly introduces the principal of PCA, LSTM, and normalization. Section 3 expounds the system of two-stage LSTM with classified PCA (CPCA) model in detail. Section 4 introduces the application of two-stage LSTM with CPCA model in S&P 500 data. Also, this section presents a series of experiments to test the performance of proposed model. Section 5 concludes this paper and gives some brief comments.

2. Methodology

In this section, several basic concepts and models used in the developed hybrid forecasting method are described as follows.

2.1. Normalization

Normalize the input features is extremely important since the features are not in the same scale which highly possible let the features with big numeric ranges dominate features with small numeric ranges. In this paper, the Minmax scaler is utilized to normalize the input features.

Let the matrix of m features in D be

$$\mathrm{D} = \begin{bmatrix} d_{11} & d_{21} & \cdots & d_{m1} \\ d_{12} & d_{22} & \cdots & d_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1n} & d_{2n} & \cdots & d_{mn} \end{bmatrix} \tag{1}$$

The normalized input features are calculated as follows:

$$\widehat{D}_{i,j} = \frac{D_{i,j} - D_{\min j}}{D_{\max j} - D_{\min j}} \tag{2}$$

where m represents the number of data features and n represents the number of each data. $D_{i,j}$ is the value of j^{th} feature on i^{th} month, $\widehat{D}_{i,j}$ is the normalized value of $D_{i,j}$, $D_{\min j}$ is the minimum value of j^{th} feature, and $D_{\max j}$ is the maximum value of j^{th} feature.

2.2. Principal component analysis

The principal component analysis (PCA) is a widely used dimensionality reduction algorithm (Chen and Hao, 2018). It is used to convert n-dimensional data to k-dimensional data (n>k>0).

Suppose there are m n-dimension data. Firstly, organize the raw data into n rows and m columns matrix F which is Eq. 1.

$$\begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_R \end{pmatrix} (a_1\ a_2\ \cdots\ a_m) = \begin{pmatrix} p_1 a_1 & \cdots & p_1 a_m \\ \vdots & \ddots & \vdots \\ p_R a_1 & \cdots & p_R a_m \end{pmatrix} \tag{3}$$

where p_i is a row vector, representing the i^{th} basis, a_j is a y_i vector, representing the j^{th} original data record.

Secondly, every row of D is going to be zeroed, which means that we are going to subtract the mean of that row.

Thirdly, calculate the covariance matrix. Mathematically, the correlation can be expressed by the covariance of two fields, then can be calculated as follows:

DOI: 10.24818/18423264/54.3.20.06

$$\mathrm{Cov(a,b)} = \frac{1}{m}\sum_{i=1}^{m} a_i b_i \quad (4)$$

The result of converting the raw data into the set of bases is that the covariance is equal to 0, and the variance is big enough. The covariance matrix is:

$$\mathrm{C} = \frac{1}{m}\mathrm{X}X^T \quad (5)$$

Fourthly, calculate the eigenvalues and corresponding eigenvectors of the covariance matrix. Suppose C is the covariance matrix corresponding to the original data matrix X, and P is a matrix composed of bases in rows, and set Y = PX. Suppose the covariance matrix of Y is D, then the relationship between D and C can be derived:

$$\mathrm{D} = \frac{1}{m}YY^T = \frac{1}{m}(PX)(PX)^T = \frac{1}{m}PXX^TP^T = P\left(\frac{1}{m}XX^T\right)P^T = PCP^T \quad (6)$$

The goal of optimization is to find a matrix P, which satisfies PCP^T as a diagonal matrix. A real symmetric matrix with n rows and n columns must find n unit orthogonal eigenvectors which expressed as

$$\mathrm{E} = (e_1\ e_2\ \cdots e_n) \text{ (put it in columns)} \quad (7)$$

The covariance matrix is shown as follows where λ represents eigenvector:

$$E^TCE = \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} (\Lambda \text{ is a diagonal matrix}). \quad (8)$$

From last step we can conclude that:

$$\mathrm{P} = E^T \quad (9)$$

Finally, the conclusion is that Y = PX is the data reduced to k dimension.

2.3. Long short-term memory network

As a special kind of RNN, the LSTM has a special design called gate which contains input gate, forgotten gate and output gate.

The input of the input gate is a linear combination of the input vectors, and the output is the operation result of the function expression. The input and output are as shown:

$$x_\rho^t = \sum_{i=1}^{I} w_{ip}\, x_i^t + \sum_{h=1}^{H} w_{h\rho}\, y_h^{t-1} \quad (10)$$

$$y_\rho^t = f(x_\rho^t) \quad (11)$$

where x is input of the cell and y is output of the cell. t can be considered as the serial number of each LSTM. Here, $w_{h\rho}$ can be seen the weight of the activation function.

The first step of the LSTM is to determine what information should be thrown away from the cellular state. This determination is made by the sigmoid function in forget gate. This gate reads h_{t-1} and x_t and then a value between 0 and 1 for each number in cell state C_{t-1} can be acquired.

As can be seen in the Fig.1, h_{t-1} represents last cell's output, x_t represents present cell's input.

DOI: 10.24818/18423264/54.3.20.06

The second step of the LSTM is to determine how much new information to add to the cell state. Next input door is divided into two parts. Sigmoid function is applied in the first part to deal with x_t to get result. Tanh function is utilized in the second part to deal with x_t to get another result. Multiply the two results to update cell status. The input of the output gate comes from the output layer and the memory block, and the input method is shown：

$$x_\pi^t = \sum_{i=1}^{I} w_{i\pi} x_i^t + \sum_{h=1}^{H} w_{h\pi}\, y_h^t \tag{12}$$

The third step of the LSTM is output door. It determines the output information h_t and the output cell status. The formula of the forgotten gate is similar to the input and output gate, and its input and output are as follows:

$$x_\emptyset^t = \sum_{i=1}^{I} w_{i\emptyset} x_i^t + \sum_{h=1}^{H} w_{h\emptyset}\, y_h^{t-1} \tag{13}$$

$$y_\emptyset^t = f(x_\emptyset^t) \tag{14}$$

LSTM has several superior features such as powerful distributed storage and robustness which can deal with complex nonlinear relationships.

3. Proposed Model

In this part, a prediction model is designed based on Two-Stage LSTM networks to perform predictions according to S&P price. The model is trained based on the top 70 percent of the historic S&P price and then 30 percent is used for performing prediction.

3.1. Classification

In this paper, the datasets used for this study is composed of 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume', 'MA', 'RSI', 'ON BAR VOL', 'CHOPPINESS', 'HIST VOL', 'Dividend Yield', 'PE Ratio', 'Earnings Yield', 'Inflation Adjusted'.

Let S be the matrix of close price and related factors, it could be described as follows:

$$\mathrm{S} = \begin{bmatrix} p_1 & f_{11} & f_{21} & \cdots & f_{m1} \\ p_2 & f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ p_n & f_{1n} & f_{2n} & \cdots & f_{mn} \end{bmatrix} \tag{15}$$

where p_i, $f_{1\,i}, f_{2\,i}, f_{3\,i} \ldots f_{\mathrm{m}\,i}$ $(\mathrm{i} = 1, 2, \ldots, 15)$ represent the close price and other factors of i^{th} month respectively. To generate the class label, the change rate of close price is applied to classify the dataset.

$$R_i = \frac{p_{i+1} - p_i}{p_i}, \forall \mathrm{i} = 1, 2, 3, \ldots, \mathrm{n} - 1 \tag{16}$$

where R_i is the changing rate of the close prices from i^{th} month to $(i+1)^{th}$ month. The class is labeled as follows:

$$y_i = \begin{cases} 1 & if \quad \Delta R_i \le \epsilon_1 \\ 2 & if \quad \epsilon_1 < \Delta R_i \le \epsilon_2 \\ 3 & if \quad \Delta R_i \ge \epsilon_2 \end{cases} \tag{17}$$

DOI: 10.24818/18423264/54.3.20.06

where ϵ_1, ϵ_2 is the low threshold value, high threshold value respectively, chosen depending R_i. For the balance of three classes, each class is assigned about one-thrid datasets after sorting the R_i sequence from small to large as follows:

$$\begin{array}{cccccc} unsorted: & R_1 & R_2 & R_3 & \cdots & R_n \\ sorted: & R_j & R_k & R_t & \cdots & R_q' \end{array} \quad (1 \le j,k,t,q \le n \text{ and } R_j \le R_k \le R_t \le R_q) \tag{18}$$

Let Rs_i be the sorted R_i, and then the low threshold, high threshold can be descripted as follows:

$$\epsilon_1 = Rs_{\lfloor n/3 \rfloor}, \epsilon_2 = Rs_{2*\lfloor n/3 \rfloor} \tag{19}$$

Considering the currency inflation in the last several years, it is much better to select changing rate R_i to replace close price p_i in the stock prediction. The class labels y_i can be generated by utilizing R_i and now the dataset S is reformulated as follows:

$$S = \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} & R_1 & y_1 \\ f_{12} & f_{22} & \cdots & f_{m2} & R_2 & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ f_{1n} & f_{2n} & \cdots & f_{mn} & R_n & y_n \end{bmatrix} \tag{20}$$

The R_{i+T} is the dependent variable which will be predicted based on the data of former T months as follows:

$$X_{i+T} = \begin{bmatrix} R_i & R_{i+1} & \cdots & R_{i+T-1} \\ p_{1\,i} & p_{1\,i+1} & \cdots & p_{1\,i+T-1} \\ p_{2\,i} & p_{2\,i+1} & \cdots & p_{2\,i+T-1} \end{bmatrix}, (1 \le \mathrm{i} \le \mathrm{n} - \mathrm{T}, T \le n-1) \tag{21}$$

where T is the number of the former months used for prediction.

Then the S can be reformulated as follows:

$$S = \left[\begin{array}{c|cc} X_{1+T} & R_{1+T} & y_{1+T} \\ X_{2+T} & R_{2+T} & y_{2+T} \\ \vdots & \vdots & \vdots \\ X_n & R_n & y_n \end{array}\right], (1 \le T \le n-1) \tag{22}$$

Separate the dataset S into three independent parts using the label of y_i.

$$S_1 = \{S|y_i = 1\}, S_2 = \{S|y_i = 2\}, S_3 = \{S|y_i = 3\} \tag{23}$$

where S_1 is the dataset when $y_i = 1$, S_2 is the dataset when $y_i = 2$, and S_3 is the dataset when $y_i = 3$.

3.2. A prediction model based on Two-Stage LSTM

The main purpose of the work is to predict the increase rate R of the next month based on the given historical increase rate R and five principle components p_1, p_2, p_3, p_4, p_5. The increase rate R_{i+T} of $(i+T)^{th}$ month is predicted based on the matrix X_{i+T} defined above.

Where T represents the selection of the size is T months. The predicted increase rate $\hat{R}_{i+T}$ is defined as follows:

DOI: 10.24818/18423264/54.3.20.06

$$\hat{R}_{i+T} = F(X_{i+T}) \tag{24}$$

$F(.)$ is the function we are going to learn later.

3.2.1. First stage LSTM model

Firstly, use $LSTM_0$ to predict the label y_{i+1} of the $(i+1)^{th}$ month based on the former T months.

where $LSTM_0$ is a prediction model which trained with train set S with the column R_{i+T} removed.

y_{i+T} is the label of the $(i+T)^{th}$ month, T is the time window size, X_{i+T} is one matrix defined above which will be used to predict the y_{i+T}. It can be described as follows:

$$y_{i+T} = LSTM_0(X_{i+T}) \tag{25}$$

Secondly, if the predicted label $LSTM_0(X_{i+T}) = 1$, we will use $LSTM_1$ to predict the change rate R_{i+1} of the $(i+1)^{th}$ month. If the predicted label $LSTM_0(X_{i+T}) = 2$, then we use $LSTM_2$. If the predicted label $LSTM_0(X_{i+T}) = 3$, then we use $LSTM_3$.

$LSTM_1$ is a prediction model which trained with train set S_1 with the column y_{i+T} removed. $LSTM_2$ is a prediction model which trained with train set S_2 with the column y_{i+T} removed. $LSTM_3$ is a prediction model which trained with train set S_3 with the column y_{i+T} removed.

The model could be represented as follows:

$$\hat{R}_{i+T} = F(X_{i+T}) = \begin{cases} LSTM_1(X_{i+T}) & ,if\ LSTM_0(X_{i+T}) = 1 \\ LSTM_2(X_{i+T}) & ,if\ LSTM_0(X_{i+T}) = 2, \\ LSTM_3(X_{i+T}) & ,if\ LSTM_0(X_{i+T}) = 3 \end{cases} \tag{26}$$

$$(1 \le \mathrm{i} \le \mathrm{n} - \mathrm{T}, T \le n-1)$$

3.2.2. Residual error optimizer based on second stage LSTM

Let $\hat{R}$ be the matrix of predicted change rate using the above LSTM prediction model and let R be the matrix of real change rate, described as follows:

$$\hat{R} = [\hat{R}_{1+T} \quad \hat{R}_{2+T} \quad \hat{R}_{3+T} \quad \cdots \quad \hat{R}_n]^T \tag{27}$$

$$R = [R_{1+T} \quad R_{2+T} \quad R_{3+T} \quad \cdots \quad R_n]^T \tag{28}$$

And then define the residual error as RE:

$$RE = [R_{1+T} - \hat{R}_{1+T} \quad R_{2+T} - \hat{R}_{2+T} \quad R_{3+T} - \hat{R}_{3+T} \quad \cdots \quad R_n - \hat{R}_n]^T \tag{29}$$

Assuming the selected time windows size is still T, we can construct one training set SE for training one LSTM prediction model $LSTM_4$ which can predict the residual error RE of the next month.

$$SE = \begin{bmatrix} RE_1 & RE_2 & \cdots & RE_T \\ RE_2 & RE_3 & \cdots & RE_{T+1} \\ \vdots & \vdots & \ddots & \vdots \\ RE_{n-T} & RE_{n-T+1} & \cdots & RE_{n-1} \end{bmatrix} \begin{bmatrix} RE_{T+1} \\ RE_{T+2} \\ \vdots \\ RE_n \end{bmatrix}, T \le n-1 \tag{30}$$

$$\widehat{RE}_{T+i} = LSTM_4(SE_{T+i}) \tag{31}$$

where $\widehat{RE}_{T+i}$ means the predicted residual error of $(T+i)^{th}$ month, and the SE_{T+i} means the data of $(T+i)^{th}$ row with RE_{T+i} removed.

DOI: 10.24818/18423264/54.3.20.06

The optimized change rate could be denoted as $\check{R}$, and is calculated as follows:

$$\check{R} = \hat{R} - RE + \widehat{RE} \tag{32}$$

And it is easy to convert optimized change rate $\check{R}$ to $\hat{p}$ as follows:

$$\hat{p}_{i+1} = \hat{R}_{i+1} * p_i + p_i \tag{33}$$

where $\hat{p}$ is the predicted close price, $\hat{p}_{i+1}$ is the predicted close price of $(i+1)^{th}$ month, and p_i is close price of i^{th} month.

3.3. Two-stage LSTM with CPCA forecasting model

This section expatiates on Two-Stage LSTM with CPCA model for stock price prediction and analysis. As illustrated in Fig. 1, four basic steps are included in this framework and they are described as follows:

Step 1: Original monthly data such as opening price, the highest price, the PE ratio are collected and stored into a matrix which can be seen in Eq. 3. In this phase, a new index R_i is put forward, which is defined in Eq. 20.

Step 2: PCA and classification are utilized to do data pre-processing for the collected data set. Using PCA to integrate the data in phase (a) into five comprehensive indicators which is the most effective processing method of multi-factor data. Stock price growth rates are regarded as a basis to classify the data after PCA. The principle of calculation can be found in Section 3.1.

Step 3: In the two-stage LSTM model, the data divided in phase (b) are predicted separately as the model prediction in the first stage. The main function of the second stage LSTM is residual correction. The residual correction of the three classification can obtain more accurate prediction effect.

Step 4: Three comparative experiments are conducted. The first comparison group is between two-stage LSTM with PCA and PCA-LSTM. This comparison is to state the effect of residual correction on the model. The second comparison is between PCA-LSTM and deep learning. It can illustrate the effect of pre-processing multifactor data and deep learning. The final comparison is between Two-Stage LSTM with CPCA model and two-stage LSTM with PCA model. This comparison is to expatiate the great significance of classification.

4. Experiment & Comparison

4.1. Data collection

In this paper, datasets are selected from S&P 500, which are average value among S&P 500. Table1 describes the collected these monthly stocks' average value datasets in thirty years spanning from April 1st in 1990 to June 1st in 2019. Yahoo Finance website (https://finance.yahoo.com/), as a publicly available data source, is the origin of the data used in this article. Fig. 2 is an overview of the original data. Utilizing test datasets, a new index R_i is proposed Eq. 16.

In this study, the proposed prediction model is carried out by using Python3.7.4 and PyCharm 2019.1.3. Python and PyCharm are used to process the raw data and construct LSTM model.

DOI: 10.24818/18423264/54.3.20.06

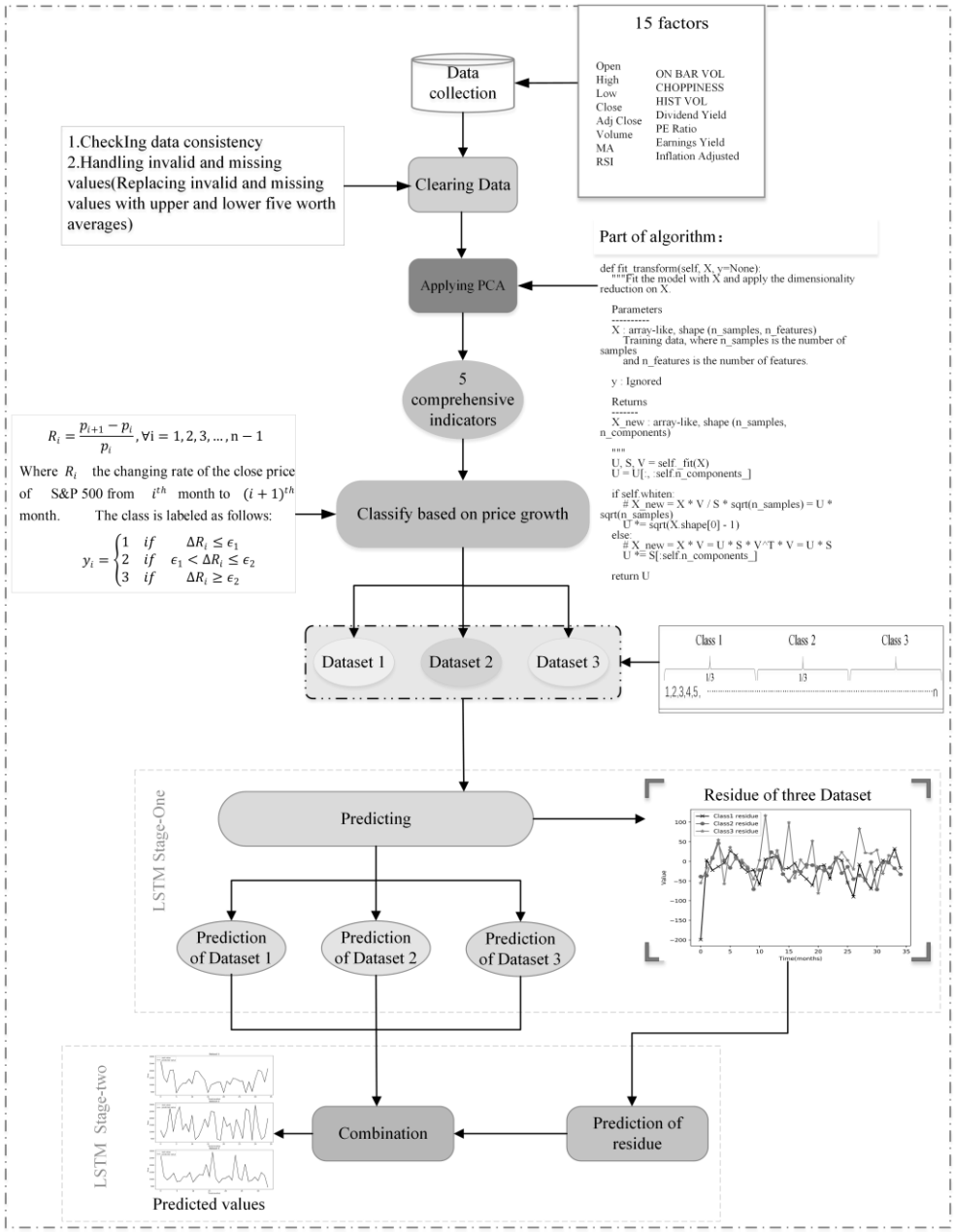

Figure 1. The framework of the developed model

Table 1. The collection of influence factors

Influence factor	Influence factor
S&P 500 Close Price	S&P 500 Low price
S&P 500 Price Earnings (PE) Ratio	S&P 500 Adjusted Closing Price (Adj

DOI: 10.24818/18423264/54.3.20.06

	Close)
S&P 500 Earnings Yield (EY)	S&P 500 Volume
S&P 500 Inflation Adjusted	S&P 500 Moving Average (MA)
S&P 500 Dividend Yield	S&P 500 Relative Strength Index (RSI)
S&P 500 Open price	S&P 500 Barco Volume (BAR VOL)
S&P 500 High price	S&P 500 Choppiness
S&P 500 Historical Volume (Hist vol)	

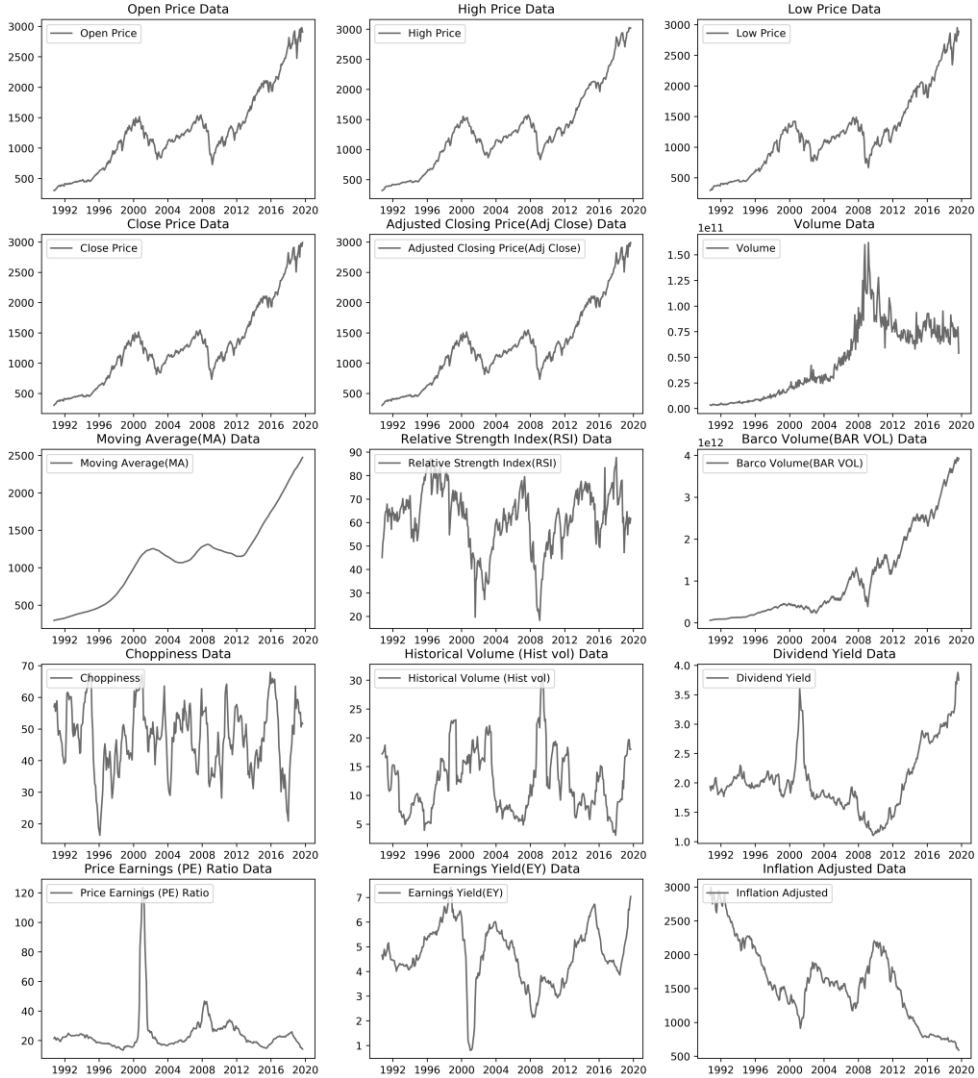

Figure 2. Data Collection

DOI: 10.24818/18423264/54.3.20.06

4.2. Apply PCA and Ratio Classify

In stock market, several factors have effect on stock price. Correlation is unavoidable among influence factors. Under this situation, the stock price is complicated and changeable which is hard to calculate. To solve above problems, PCA is a valid method. In this experiment, fifteen influence factors are chosen and listed in Table 1. Finally, five principle components are synthesized by calling sklearn.decomposition in python.

From Section 3 we know the definition of growth rate of stock price R_i. Then, arrange the data in order from smallest to largest. Take the first $1/3$ as the first dataset, take the second $1/3$ as the second dataset, and the rest is the third dataset. Price classification reduces the error caused by currency depreciation which improves the science and accuracy of prediction.

After completing the above two steps, the data that is processed by PCA and the data after the classification correspond to the time order and enter into the LSTM model for training and testing.

4.3. Model evaluation metric

Coefficient of determination (R^2) represents the reliability of prediction in regression model. The closer R^2 to 1, the better model performance. Root mean square error (RMSE) is utilized to measure the deviation between the observed value and the real value. Mean Absolute Percentage Error (MAPE) is an absolute error calculation method. The formulas are as follows in Table 2, where Z_α represents the size of test data. y_i and Y_i are the value of prediction and original data, respectively.

Table 2. Model evaluation metric

Metric	Formula
R^2	$R^2 = 1 - \frac{\sum_{i=1}^{Z_a}(y_i - Y_i)^2}{\sum_{i=1}^{Z_a}(Y_i - \bar{Y}_i)^2}$
RMSE	$RMSE = \sqrt{\frac{1}{Z_a}\sum_{i=1}^{Z_a}(y_i - Y_i)^2}$
MAE	$MAE = \frac{1}{Z_a}\sum_{i=1}^{Z_a}\lvert y_i - Y_i\rvert$
MAPE	$MAPE = \frac{100\%}{Z_a}\sum_{i=1}^{Z_a}\frac{\lvert y_i - Y_i\rvert}{Y_i}$

4.3. Trading simulation and comparisons

Specific principle contents of the model have been introduced in Section 3. In Two-Stage LSTM model, there are differences in the parameter settings between two stage. Different from the traditional LSTM model, LSTM is used in all five hidden layers. It is used to improve the accuracy to achieve the purpose of improving the prediction effect. Each layer has a different number of neurons: The

DOI: 10.24818/18423264/54.3.20.06

first LSTM has 100 neurons, the second LSTM has 80 neurons, the third LSTM has 40 neurons, the fourth LSTM has 20 neurons and the last LSTM has 1 neuron. In this stage, the number of epochs is 1000 and batch size equals to 32. In the second stage, four-tier LSTM in a LSTM shows the best result. The number of neurons from the first layer to the fourth layer are respectively 80,60,40,1 and the batch size is 28 with 500 epochs. The default activation function 'tanh' is used both in the first stage and the second stage. The first 70% of the dataset are selected as the training set and the last 30% as the test set. Tables 3-4 lists the comparison among two-stage LSTM with CPCA, two-stage LSTM with PCA, PCA-LSTM and BP while the same testing set are applied. In this section, the results of the above methods will be compared and the comparison results are demonstrated in the Tables 3-4.

Table 3. The performance of two-stage LSTM with CPCA

	R^2	RMSE	MAE	MAPE
Class 1	0.992741	46.636272	29.568098	2.554785%
Class 2	0.998661	29.868677	23.941577	1.958978%
Class 3	0.994633	42.660839	32.026552	2.853410%
Mean	0.995345	39.721929	28.512076	2.455724%

Table 4. The performance of compared models

Model	R^2	RMSE	MAE	MAPE
BP	0.849745	200.979889	178.772056	8.663114%
PCA-LSTM	0.978965	75.543425	57.488045	2.773953%
Two-stage LSTM with PCA	0.979379	74.795040	55.811595	2.711066%

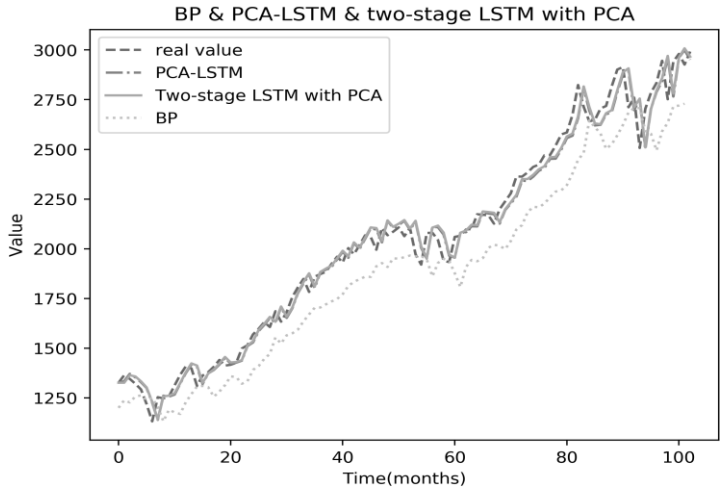

Figure 3. Prediction comparisons of different models

DOI: 10.24818/18423264/54.3.20.06

In this paper, a new deep learning model is proposed to forecast stock price. Considering the complexity of factors that affect stock prices, PCA is selected to synthesize the data into five independent comprehensive indices. In order to make the input model data more scientific and effective, $R_i = \frac{p_{i+1}-p_i}{p_i}$ is used to classify the original data. Then, data is trained and tested by two-stage LSTM after applying PCA and classification. Two-stage LSTM model means that both prediction and residual analysis utilize LSTM for prediction and analysis.

(a) Comparison between BP and PCA-LSTM

From Table 4, it is vividly shown that every indicators of PCA-LSTM perform better than BP. The R^2 of PCA-LSTM is 0.9790 which is much higher than BP. The comparison results are more intuitive in Fig. 3. In Fig. 3, the predicted value of BP fluctuates greatly and differs greatly from the real value, which well explains the value of RMSE, MAE and MAPE in Table 4. This result shows that deep learning such as LSTM has better prediction than shallow neural network like BP.

(b) Comparison between PCA-LSTM and two-stage LSTM with PCA

In this set of comparisons, the results of two models show that residual correction play a very important role in the prediction model. Other things being equal, the fitting value of Two-Stage LSTM with PCA is 0.9794 while PCA-LSTM is 0.9790. From Table 3, it is obviously to see that the MAE and RMSE of PCA-LSTM are 57.4880 and 75.5434 while the MAE and RMSE of two-stage LSTM with PCA are 55.8116 and 74.7950. The predicted result of PCA-LSTM and two-stage LSTM with PCA can be seen in Fig. 3.

(c) Comparison between two-stage LSTM with PCA and two-stage LSTM with CPCA

The classified data models are presented in Table 3. The fitting values of the three groups of classified data are all greater than 0.9794. This is higher than the R^2 of Two-Stage LSTM with PCA. It is also obvious from the numerical results of RMSE. The average of RMSE value in the two-stage LSTM with CPCA is 39.7219 which is smaller than that in two-stage LSTM with PCA. Also, the degree of fitting between the predicted value and the real value of the three categories in the two-stage LSTM with CPCA is quite higher than that in two-stage LSTM with PCA by comparing Table 3 and Table 4.

Through the comparison experiment of the above three groups, the effectiveness of two-stage LSTM with CPCA can be fully proved. The model put

DOI: 10.24818/18423264/54.3.20.06

forward in this paper has high prediction accuracy and practical significance in the actual stock price measurement.

5. Conclusions

In this paper, an efficient model which integrates PCA with data classification into two-stage LSTM is proposed to predict the stock price. The first step is to clear the original data by using PCA and classification. PCA is utilized to reduce the dimension in order and simplify the computational complexity. Then divide the dataset into three categories on the basis of stock price growth rate. The second step is to input the data processed by PCA and classification into LSTM model with residual correction for training and testing. Finally, three comparative experiments are designed and the result verifies the validity of the model.

In future studies, the established model needs further improved to achieve better prediction. Firstly, an alternative approach to PCA is a starting point. Other methods of multi-factor data processing will be studied so as to find the best data pre-processing method. Secondly, alternative models may be considered to replace the residual correction LSTM model. For instance, Markov Chain is an option. Through further research and improvement of the model, better prediction results are obtained step by step.

ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China (Grant No. 71971122 and 71501101) and the Natural Science Foundation of Jiangsu Province (Grant No. BK20150928).

REFERENCES

[1] **Roy, S.S., Chopra, R., Lee, K.C., Spampinato, C., Mohammadi-Ivatlood, B. (2020),** ***Random Forest, Gradient Boosted Machines and Deep Neural Network for Stock Price Forecasting: A Comparative Analysis on South Korean Companies****. International Journal of ad Hoc and Ubiquitous Computing, 33(1): 62-71;*

[2] **Lin, Z.(2018),** ***Modelling and forecasting the Stock Market Volatility of SSE Composite Index Using GARCH Models****. Future Generation Computer Systems, 79: 960-972;*

[3] **Baffour, A.A., Feng, J.C., Taylor, E.K.(2019),** ***A Hybrid Artificial Neural Network-GJR Modeling Approach to Forecasting Currency Exchange Rate Volatility****. Neurocomputing, 365: 285-301;*

[4] **Lee, T.K., Cho, J.H., Kwon, D.S., Sohn, S.Y. (2019),** ***Global Stock Market Investment Strategies Based on Financial Network Indicators Using Machine Learning Techniques****. Expert Systems with Applications, 117: 228-242;*

DOI: 10.24818/18423264/54.3.20.06

[5] **Khashei, M., Hajirahimi, Z. (2019),** ***A Comparative Study of Series Arima/Mlp Hybrid Models for Stock Price Forecasting****. Communications in Statistics-Simulation and Computation, 48(9): 2625-2640;*
[6] **Vilela, L.F.S., Leme, R.C., Pinheiro, C.A.M., Carpinteiro, O.A.S. (2019),** ***Forecasting Financial Series Using Clustering Methods and Support Vector Regression****. Artificial Intelligence Review, 52(2): 743-773;*
[7] **Gong, X.L., Liu, X.H., Xiong, X., Zhuang, X.T. (2019),*****Forecasting Stock Volatility Process Using Improved Least Square Support Vector Machine Approach****. Soft Computing, 23(22): 11867-11881;*
[8] **Lahmiri, S.(2018),** ***Minute-ahead Stock Price Forecasting Based on Singular Spectrum Analysis and Support Vector Regression****. Applied Mathematics and Computation, 320: 444-451;*
[9] **Xu, W.Q., Peng, H., Zeng, X.Y., Zhou, F., Tian, X.Y., Peng, X.Y. (2019),** ***A Hybrid Modelling Method for Time Series Forecasting Based on a Linear Regression Model and Deep Learning****. Applied Intelligence, 49(8): 3002-3015;*
[10] **Liu, Y. (2019),** ***Novel Volatility Forecasting Using Deep Learning-Long Short Term Memory Recurrent Neural Networks****. Expert Systems with Applications, 132: 99-109;*
[11] **Sagheer, A., Kotb, M. (2019),** ***Time Series Forecasting of Petroleum Production Using Deep LSTM Recurrent Networks****. Neurocomputing, 323: 203-213;*
[12] **Zhang, J.H., Yan, J., Infield, D., Liu, Y.Q., Lien, F.S. (2019),** ***Short-term Forecasting and Uncertainty Analysis of Wind Turbine Power Based on Long Short-Term Memory Network and Gaussian Mixture Model****. Applied Energy, 241: 229-244;*
[13] **Rufino, M.M., Bez, N., Brind'Amour, A. (2019),** ***Influence of Data Pre-processing on the Behavior of Spatial Indicators****. Ecological Indicators, 99: 108-117;*
[14] **Henrique, B.M., Sobreiro, V.A., Kimura, H. (2018),** ***Stock Price Prediction Using Support Vector Regression on Daily and up to the Minute Prices****. The Journal of Finance and Data Science, 4: 183-201;*
[15] **Zhong, X., Enke, D. (2017),** ***Forecasting Daily Stock Market Return Using Dimensionality Reduction****. Expert Systems with Applications, 67: 126–139;*
[16] **Wang, J., Wang, J. (2015),** ***Forecasting Stock Market Indexes Using Principle Component Analysis and Stochastic Time Effective Neural Networks****. Neurocomputing, 156: 68-78;*
[17] **Ouyang, T.H., Zha, X.M., Qin, L., He, Y.S., Tang, Z.H.(2019),** ***Prediction of Wind Power Ramp Events Based on Residual Correction****. Renewable Energy, 136: 781-792*;
[18] **Chen, Y.J., Hao, Y.J. (2018),** ***Integrating Principle Component Analysis and Weighted Support Vector Machine for Stock Trading Signals Prediction****. Neurocomputing, 321: 381-402.*

DOI: 10.24818/18423264/54.3.20.06